HCAL Equip DB

HCAL_WEDGE: define a wedge by its ETA, PHI (Max 16) and layer (max 17). Each wedge has WEDGE_NAME and assigned a sequence number WEDGE_ID as primary key.

HCAL_RBX & HCAL_RBX_HIST: hcal_rbx holds most current version of all the RBX's. RBX_ID is sequence number as a primary key. Each RBX is assigned such ID. CREATE_TIME is the time when a RBX is recorded into db and the MODIFY_TIME is the same as the CREATE_TIME at the first time, then changes when the record is modified. The initial RBX_VERSION is 0, incremented by one when repaired. STATUS describes if a RBX is in-use, broken, or spare. Barcode (20 byte varchar2) uniquely identifies RBX unit physically. HISTORY is a record to what have been happen to a RBX. When a new RBX registers to the db, First, we insert into HCAL_RBX db with the CREATE_TIME and CREATOR same as MODIFY_TIME and MODIFIER, then into HCAL_RBX_HIST with the current time as the TIMESTAMP. CREATOR, MODIFIER, BOOKKEEPER attributes may be the user name of the unix account or oracle account. This depends how the account is set up. When a record is changed, we update the HCAL_RBX, then insert new record into HCAL_RBX_HIST. A RBX may change its version, status, location (wedge) or history. All the time related attributes here are unix timestamps that are integers with length of 10.

HCAL_RM/HCAL_RM_HIST: RM has all the common attributes in RBX. One RBX has four different RM's. So there is a TYPE attribute to identify the four different RM's. Each type of RM has its own position in RBX and not exchangeable. One RM can change its version, status, history, and the RBX that it may belong to at certain time period.

HCAL_WEDGE_CONNECTOR: a HCAL wedge has 4 PHI's, each PHI has 17 connectors reflecting 17 layers, each connector has 16 optical fibers go through for the 16 ETA's. So a WEDGE_ID, PHI (1 to 4) and LAYER(1 to 17) uniquely identify a connector. There are 4x17 connectors/wedge. CONN_ID is a sequence number as primary key. CONN_NUM is the connector label.

HCAL_WEDGE_ODU_CONNECTION: every time, a new connection created the CONN_ID is increased by 1. There are six dimensions to identify a connection. They are:

1. ODU_ID,
2. ODU_conn_num (1-17) that are 17 connectors/ODU,
3. WEDGE_ID,
4. PHI (1-4) that are 4 PHI's/wedge,
5. layer (1-17) that are 17 layers/wedge,
6. CABLE_ID,

STATUS attribute is used when a new one replace an old one that will be marked as 'retired'. The retiring time and person who changed the status to retire are recorded into RETIRE_TIME and STATUS_MODIFIER.

HCAL_ODU: HCAL_ODU has all the common attributes in HCAL_RBX. One RM may have only one ODU at any given time. Any ODU may only belong to one RM at any given time. An ODU can change its version, status, history and the RM that it belongs to.

HCAL_HPD: similar to HCAL_ODU.

HCAL_FE_CARD/HCAL_FE_CARD_HIST: HCAL_FE_CARD has all the common attributes in HCAL_RBX. One RM may have three FE card and one FE may have six QIE card. So there are 18 QIE cards per RM. One FE card can only belongs to one RM at any given time. A FE card can change its version, status, history, position (described by the slot attribute, integer: 1-3).

HCAL_SUBRM_HIST: It holds all history of ODU, HPD, and FE card. The slot attribute is optional. So for ODU and HPD, there is noting needed because there is only one ODU and HPD for each RM., but for FE card, it is 1 to 3.

HCAL_physical_location: It contains all the locations where HCAL RACK will be place. Possible location names are: UX5 (Collision Hall), US5 (counting house), SX5 (Serface Building), H2Floor (Detector Area), H2Control (Control/counting room), Spare Room.

HCAL_RACK/HCAL_RACK_HIST/HCAL_CRATE/HCAL_CRATE_HIST: They have all the common attributes. One RACK can have multiple Crates. There are a lot of attributes gotten from CMS EMDB which are changing. We don't know exact what the format, value they are and how they change with time. We need more info from EMDB people.

HCAL_HTR/HCAL_HTR_HIST: They have all the common attributes. Slot is used for the HTR location in a crate. There are 18 HTRs per HCAL readout crate. So the slot is from 1 to 18.

HCAL_FE_HTR_CONNECTION: there are two connectors per FE and 3 fibers per FE connector. X connectors per HTR and Y fibers per HTR connector. One cable bundle (which has barcode) has Z fibers grouped within Z1 connectors. Similar to **HCAL_WEDGE_ODU_CONNECTION**, when a new connection created, CONN_ID increases by 1. The old connection retired with the STATUS marked. Any of the following attributes changed will create a new connection: QIE_ID, FE_CONN_NUM(1-2), HTR_ID, HTR_CONN_NUM(1-X), and CABLE_ID.

HCAL_DCC/HCAL_DCC_HIST: They have all the comment attributes. One DCC per HCAL crate.

HCAL_CPU/HCAL_CPU_HIST: They have all the comment attributes and IP address and CPU_NAME that may change with time. One CPU per HCAL crate.

User cases:
If a RBX (say RBX_ID1) is broken and a new one (RBX_ID100) will replace the broken one. This is operated by USERX. Here is what will go through in the db:

1. Insert into HCAL_RBX_HISTS with (RBX_ID1, timestamp1, NULL, USERX, "broken", 0, "need to be repaired")
2. Insert into HCAL_RBX_HISTS with (RBX_ID100, timestamp1, the wedge_id used by RBX_ID1, USERX, "in use", 0, "taken online")
3. Update HCAL_RBX with MODIFY_TIME=timestamp1, modifier=USERX, status="broken", wedge_id=null, history="to be repaired" and the rests are not changed.
4. Update HCAL_RBX with MODIFY_TIME=timestamp1, modifier=USERX, status="in use", wedge_id=the previous one used by RBX_ID1.
5. Insert four records into HCAL_RM_HISTS where RBX_ID=RBX_ID1 and status="removed" and the rest are the same as the previous one.
6. Insert four records into HCAL_RM_HISTS where RBX_ID=RBX_ID100 and status="in use" and rest are the same as the previous one.
7. Update four records in HCAL_RM where RBX_ID=RBX_ID1 and status="remove", MODIFY_TIME, MODIFIER and rest are the same. Find out what RM_ID's belong to the RBX_ID1. Let's say RM_ID1, RM_ID2, RM_ID3, RM_ID4.
8. Update four records in HCAL_RM where RBX_ID=RBX_ID100 and STATUS="in-use", MODIFY_TIME, MODIFIER and rest are the same. Find out what the RM_ID is. Let's say RM_ID104, RM_ID101, RM_ID102, RM_ID103.
9. Insert into HCAL_SUBRM_HIST where RM_ID=RM_ID1, … RM_ID=RM_ID4 for ODU's, HPD's and FE's.
10. Insert into HCAL_SUBRM_HIST where RM_ID_RM_ID101, … RM_ID=RM_ID104 for ODU's, HPD's and FE's.
11. Update HCAL_ODU, HCAL_HPD and HCAL_FE_CARD where RM_ID=RM_ID1, … RM_ID=RM_ID4, and find out all the ODU_ID's , say ODU_ID11, ODU_ID12, ODU_ID13 and ODU_ID14. Find out all the FE_ID's, say FE_ID11, FE_ID12, FE_ID13, FE_ID14, FE_ID15, FE_ID16, FE_ID17, FE_ID18, … ID22.
12. Update HCAL_ODU, HCAL_HPD and HCAL_FECARD where RM_ID=RM_ID100, … RM_ID104 and find out all the ODU_ID's , say ODU_ID101, ODU_ID102, ODU_ID103 and ODU_ID104. Find out all the FE_ID's , say FE_ID101, FE_ID102, FE_ID103, FE_ID104, FE_ID105, FE_ID106, …FE_ID112.

13. Insert into HCAL_WEDGE_ODU_CONNECTION with new CONN_ID and all the other info where ODU_ID=ODU_ID101, … ODU_ID=ODU_ID104
14. Update HCAL_WEDGE_ODU_CONNECTION with the status=”retired” where ODU_ID=ODU_ID10 … ODU_ID=ODU_ID14.
15. Insert into HCAL_QIE_HTR_CONNECTION with new CONN_ID and all other info where QIE_ID=QIE_ID101 … QIE_ID=QIE_ID112.
16. Update HCAL_QIE_HTR_CONNECTION with the status=”retired” where QIE_ID=QIE_ID10 … QIE_ID=QIE_ID22

What is the next:

We need info from CMS EMDB for the crate and rack attributes.
We need info on individual units such as HPD, ODU, FE_CARD, HTR and so on in order to store completed record of the units.
We need info on the measurements for individual units so we can create entities to store them.